Statistical Methods in Artificial Intelligence

Assignment 5

Deadline : 21 November 2023 11:55 P.M

Instructor: Prof Ravi Kiran Sarvadevabhatla

November 20, 2023

1 General Instructions

- Your assignment must be implemented in Python.
- Submit your assignment as a Jupyter Notebook file (.ipynb) for each question. You can have other files as you want, but please ensure the notebook is named 1.ipynb and 2.ipynb respectively.
- While you're allowed to use ChatGPT for assistance, you must explicitly declare in comments the prompts you used and indicate which parts of the code were generated with the help of ChatGPT.
- Plagiarism will only be taken into consideration for code that is not generated by ChatGPT. Any code generated with the assistance of ChatGPT should be considered as a resource, similar to using a textbook or online tutorial.
- The difficulty of your viva or assessment will be determined by the percentage of code in your assignment that is not attributed to ChatGPT. If during the viva you are unable to explain any part of the code, that code will be considered as plagiarized.
- Clearly label and organize your code, including comments that explain the purpose of each section and key steps in your implementation.
- Properly document your code and include explanations for any non-trivial algorithms or techniques you employ.
- Ensure that your Jupyter Notebook is well-structured, with headings, subheadings, and explanations as necessary.
- Your assignment will be evaluated not only based on correctness but also on the quality of code, the clarity of explanations, and the extent to which you've understood and applied the concepts covered in the course.

- Make sure to test your code thoroughly before submission to avoid any runtime errors or unexpected behavior.
- The Deadline will not be extended.
- Moss will be run on all submissions along with checking against online resources.
- We are aware of how easy it is to write code now in the presence of ChatGPT and Github Co-Pilot, but we strongly encourage you to write the code yourself.
- We are aware of the possibility of submitting the assignment late in github classrooms using various hacks. Note that we will have measures in place for that and anyone caught attempting to do the same would be given zero in the assignment.
- The relevant data files are uploaded to the courses portal.
- github classroom invitation link

2 Kernel Density Estimation

2.1 Question 1

2.2 Task-1 : (20)

Implement Kernel Density Estimation (KDE). Your KDE should include the following functions

- allows selection from the following kernels: box, gaussian, triangular
- fit data
- select appropriate bandwidth using the pseudo-likelihood method
- evaluate the density given an input x
- visualize the data and the estimated pdf. For this you can assume that the function will be used only when input dataset is 1-D or 2-D.

The KDE should support n-dimensional input, so don't write your code assuming 1-D data.

2.3 Task-2 : (80)

The next task serves as an extension of Problem 4.2 from Assignment 2 of this course. Datasets remain the same here, as well.

Recall the bounding box problem from Assignment 2. The task was to determine the optimal horizontal and vertical Euclidean distance thresholds between bounding boxes containing words on a document page. The objective of this task is to establish connections between boxes within a paragraph while ensuring that boxes across paragraphs and columns remain unconnected. Attached are illustrative examples showcasing the desired box connections and a sample visualization of the expected output ATTACHMENT.
You have also been given the following scripts -

- Script to visualize the enclosing boxes.
- Script to visualize the connecting boxes. The input for this script is a data frame object with the following attributes.
  - ID: The ID number of the word which is in int datatype.
  - Top-Left, Bottom-Right, Top edge center, Bottom edge center, Right edge center, Left edge center: A list containing the x and y coordinates of the respective coordinates as understood by their attribute names.
  - Top box, Bottom box, Right box, Left Box - A list containing the distance and id of the nearest neighbor in the Top, Bottom, Right, and Left directions respectively. **HINT - To remove the connection make sure that the list is [-1, 0].**

Using the KDE implemented in the previous task, estimate the appropriate horizontal and vertical thresholds to apply to the distances between the boxes, such that boxes within the same paragraphs are connected and there are no connections to boxes in other paragraphs.

For any 4 images of your choice from the dataset, experiment with different bandwidths . Visualize the density estimated and thresholded document for each of the hyperparameter settings per image.
Note: If you have 4 hyperparameter configurations, you should have 12 figures for each (2 density estimation curves (horizontal and vertical distances) along with the final output image) for each hyperparameter setting

Visualize and compare the generated output for the following images given by your KDE with the best hyperparameter configuration with the output obtained from your solution in Assignment 2 Question 4.2. The images are listed below :

- 29.jpg
- 68.jpg
- 145.jpg
- 201.jpg
- 232.jpg

- 250.jpg

Note : You should visualize the output images side by side for each test image

3 Hidden Markov Models

Hidden Markov Models (HMMs) are powerful statistical models widely used in various fields such as speech recognition, bioinformatics, and finance. They are particularly effective for modeling sequences of observations with underlying hidden states. HMMs belong to the family of probabilistic graphical models and are characterized by their simplicity and versatility.

3.1 Basic Components

An HMM is composed of the following basic components:

1. **States (S):** A set of hidden states that the system can be in at any given time. These states are not directly observable.
2. **Observations (O):** A set of observable symbols or emissions associated with each hidden state.
3. **Transition Probabilities (A):** The probabilities of transitioning from one hidden state to another. Represented by a matrix A, where A_{ij} is the probability of transitioning from state i to state j.
4. **Emission Probabilities (B):** The probabilities of observing a particular symbol given a hidden state. Represented by a matrix B, where B_{ij} is the probability of emitting symbol j from state i.
5. **Initial State Probabilities (π):** The probabilities of starting in a particular hidden state. Represented by a vector π, where π_i is the probability of starting in state i.

3.2 Modeling Process

The process of modeling with an HMM involves the following steps:

1. **Initialization:** Assign initial values to the transition probabilities, emission probabilities, and initial state probabilities.
2. **Forward Algorithm:** Compute the probability of observing a sequence of symbols given the model.
3. **Backward Algorithm:** Compute the probability of being in a particular state given the observed sequence.
4. **Expectation-Maximization (EM) Training:** Adjust the model parameters to maximize the likelihood of the observed data.

3.3 Applications

HMMs find applications in a variety of domains:

- **Speech Recognition:** Modeling phonemes and words.
- **Bioinformatics:** Predicting protein structures and gene locations.
- **Finance:** Analyzing financial time series data.
- **Robotics:** Tracking the movement of objects.

3.4 Task-3

You are Chottu, a casino inspector, whose job is to investigate whether a casino is trading out a fair die for a loaded (unfair) die and scamming the customers. At the same time, you are also well-versed in Statistical Methods in Artificial Intelligence, and you realize that you can use Hidden Markov Models to solve this case. You need not implement the HMM from scratch but are allowed to use any standard library. But please make sure you are familiar with the underlying theoretical concepts.

3.4.1 Dataset

You are provided with a file named `rolls.npy`, a NumPy file containing data from 50,000 observed rolls at the casino. Each roll is represented by the number on the top face of the die. For instance, if the number 5 is recorded, it indicates that a 5 was rolled. Also, assume that the first die was a fair one.

3.4.2 Part 1 (20)

1. Train an HMM model with a 50 % train and validation split (Split it sequentially, that is the first 50 % rolls should be on the train set and rest in validation. Experiment with different emission probabilities for the loaded die and report the model with the best score. Use the validation split to evaluate the best model.
2. Find the most likely sequence of switching between the fair and loaded die.
3. Plot the generated states.
4. What problem in Hidden Markov Models does this task correspond to?

3.4.3 Part 2 (20)

1. How often do you think the casino is switching out the fair die for the loaded one and vice versa?
2. What problem in Hidden Markov Models does this task correspond to?

3.4.4 Part 3 (20)

1. How do you think the loaded die is biased
2. What problem in Hidden Markov Models does this task correspond to?

Note: Use a random seed of 13 in the entire exercise.

3.5 Task-4 : (30)

Virat and Rohit are opening the batting for India. You are following the game on the radio. The local radio station's ball-by-ball commentator speaks a language you can't understand, but you can figure out the runs scored for each ball and that both openers played the entire match.
ICC has a new rule: there are no more batsman changes for odd runs or at the end of each over. However, there's roughly 30 percent chance (you don't know the exact percentage) of strike changes at the end of each ball.
As a cricket enthusiast, you know Virat is likely to focus on singles and doubles to anchor the innings. In contrast, Rohit is more prone to taking risky shots to boost the run rate.
Dataset: You are provided with `runs.npy` which contains a sequence of length 30000 representing the runs scored by the players for each ball. The possible runs scored for a ball is 0,1,2,3,4,6. Assume there were no 5 runs scored.

1. Find the optimal transition, emission and start probability for the HMM model.
2. Choosing the best HMM based on the data, and the model information can you predict who played the first and the last ball?